Collaborative Filtering Under a Sybil Attack: Similarity Metrics do Matter!

Antoine Boutet, Florestan de Moor, Davide Frey, Rachid Guerraoui,
Anne-Marie Kermarrec, Antoine Rault

Collaborative Filtering Under a Sybil Attack: Similarity Metrics do Matter!

Antoine Boutet*, Florestan De Moor†, Davide Frey‡, Rachid Guerraoui§, Anne-Marie Kermarrec¶ and Antoine Rault§

*CITI, INSA Lyon, Inria, Univ-Lyon, France
Email: antoine.boutet@insa-lyon.fr

†Univ Rennes, ENS Rennes, France
Email: florestan.de-moor@ens-rennes.fr

‡Univ Rennes, Inria, CNRS, IRISA, France
Email: davide.frey@inria.fr

§EPFL, Lausanne, Switzerland
Email: firstname.lastname@epfl.ch

¶Mediego, Rennes, France
Email: anne-marie.kermarrec@mediego.com

***Abstract*—Recommendation systems help users identify interesting content, but they also open new privacy threats. In this paper, we deeply analyze the effect of a Sybil attack that tries to infer information on users from a user-based collaborative-filtering recommendation systems. We discuss the impact of different similarity metrics used to identity users with similar tastes in the trade-off between recommendation quality and privacy. Finally, we propose and evaluate a novel similarity metric that combines the best of both worlds: a high recommendation quality with a low prediction accuracy for the attacker. Our results, on a state-of-the-art recommendation framework and on real datasets show that existing similarity metrics exhibit a wide range of behaviors in the presence of Sybil attacks, while our new similarity metric consistently achieves the best trade-off while outperforming state-of-the-art solutions.**

I. Introduction

User-based collaborative filtering exploits the opinions of users (stored in a profile) to identify the correlations in terms of interests between users. Albeit other variants of collaborative filtering exist, user-based systems remain important due to their simplicity and their availability in a number of open-source machine learning frameworks [2]. Moreover, their focus on users makes them particularly interesting in environments where the set of items to be recommended changes frequently, for example in news or media content. The ability to scale K-nearest-neighbor (KNN) computations [16] (i.e., building the graph that captures the correlation in term of interests between users according to a similarity metric) also makes user-based CF a privileged tool to cope with this highly dynamic context.

Recent research has shown, however, that user-based collaborative filtering [12], [14] represents a preferred target for attackers that wish to learn sensitive information on users. In this paper, we investigate this issue by focusing on an attack introduced by [12]. The attacker targets user-based collaborative filtering by creating a number of fake identities (i.e., Sybil attack) to extract information about a user's profile, starting from already available auxiliary information. In the domain of cross-company recommendations, for example, the attacker may be a company that wishes to learn user preferences on items managed by its competitors.

Even if it introduced the attack, [12] did not evaluate it. In a recent paper, we carried out an analysis showing its limited effectiveness with a specific metric [18]. In this paper we partially correct the conclusions of [18] by showing that the attack is indeed effective, but that it can be counteracted by specific similarity metrics.

More precisely, we enrich the state-of-the-art on the attack of [12] with two main contributions. First we carry out a thorough evaluation with a variety of similarity metrics on multiple data sets on Mahout [2], an open-source machine-learning framework. We show that the effectiveness of the attack strongly depends on the similarity metric used to build the KNN graph. Moreover, we observe that existing similarity metrics exhibit an inherent trade-off between attack resistance and recommendation quality. Second, we start from this observation and propose a novel, generic, similarity metric that can turn existing metrics into attack-resistant ones. While our approach uses two phases, it uses them to compute similarity and therefore differs significantly from algorithms [43], [42] that compute ratings in two steps. We apply our new generic metric to *Cosine* similarity, the best performing in terms of recommendation quality, and turn it into an attack-resistant metric without hampering recommendation quality. We show that our similarity-based protection allows no more than 60% of the Sybil users to succeed in a targeted attack with an ideal neighborhood, and outperforms a state-of-the-art privacy-preserving recommender [29] on a generic attack.

II. Background

The collaborative nature of most recommendation systems [28], [15], constitutes both a strength and a weakness. On the one hand, collaborative filtering can recommend complex objects that have no easily exploitable content associated with them. On the other hand, the fact that collaborative filtering combines the opinions of different users has raised concerns about the privacy threats it poses. As the data exploited by recommenders gets more and more into the personal sphere (e.g. medical data [23]), collaborative filtering faces an inherent trade-off between accuracy and privacy [31], [36], [25].

This has led to two main lines of research: identifying potential threats and attacks, and providing attack resistance.

a) Attacks against Recommenders: We can distinguish two types of attacks on recommenders: passive, and active. In the former, the attacker simply tries to learn information about other users through legitimate means. In the latter, the attacker carries out operations that go outside the standard behavior of a user like in the attack of this paper.

BlurMe [41] and [7] present passive attacks that extract demographic information such as ethnicity or gender from the ratings in a recommender. BlurMe also proposes an obfuscation mechanism to limit the impact of such an attack. [14] shows that targeted and personalized ads contain valuable information that allows accurate reconstruction of users' interest profiles. [12] analyzes, instead, how auxiliary information, obtained from the system itself or from external sources, makes it possible to extract individual user preferences from otherwise aggregate information such as related-item lists or item-covariance matrices.

The attack we consider in this paper falls instead in the active category. In this context, [7] develops an approach that maximizes the ability to learn new information by asking users to rate specific items. Pistis [27] considers an attacker that attempts to copy the profile of the target user and proposes a mechanism that limits its impact by expressing ratings on privacy-preserving groups of items. Sybil attacks employ several fake identities like Eclipse attacks in the Bitcoin network [21]. Shilling attacks [39] adopt this approach to influence the output of the recommender, for example by biasing it towards a particular brand or product [20], [38]. The attack we consider in this paper instead uses it to implement a stronger version of the attack in [27]. This Sybil attack was initially introduced by [12] and later partially evaluated by [18] with *Cos-overlap*. Our results go beyond the partial analysis of [18], highlight the tradeoff between privacy and recommendation quality, and provide a solution to it.

b) Privacy Protection in Recommenders: Protecting collaborative filtering from the above attacks constitutes a promising research direction [33]. The first attempts to provide privacy-preserving recommenders focused on decentralized solutions based on homomorphic encryption [13], anonymization [11], or profile obfuscation [6]. In a centralized setting, [34] evaluates the feasibility of applying several data obfuscation techniques. [5] and [35] proposed injecting noise into user profiles, but they were later shown to be vulnerable to statistical attacks that filter out the random noise to reconstruct the missing information [4], [24], [26]. Moreover, all the above solutions remain vulnerable to attacks that combine recommended items with auxiliary information available through external sources, like the one we study in this paper.

Systems that apply differential privacy only to neighborhood computation [44] exhibit the same problem. But some authors have also proposed systems that incorporate randomization and ensure differential privacy when they generate recommendations [10], [32]. In [10], the authors demonstrate that their approach can effectively counteract a Sybil-based censorship attack. However, its effectiveness against an attacker equipped with external auxiliary information remains unclear.

[29] considers the same Sybil attack as studied in this paper and proposes PPNS. In [30] the same authors provide a different version of PPNS that should provide better attack resistance but lower recommendation quality. But in Section VII-F, we show that our approach significantly outperforms PPNS.

Finally, [37] proposes a reputation score that aims to count a new rating of a neighbor only if this user rated sufficiently many items in common with the targeted user. In this paper, we consider instead an adversary that clones a part of the profile of the target user. Consequently, [37] is complementary to our solution and not a concurrent approach.

III. System model

We consider a recommender based on user-based collaborative filtering, a scheme implemented in many machine-learning frameworks. For each user, u, the system maintains a user-profile data structure which collects the mapping between users, items, and the associated numerical scores (e.g. 1 to 5). A user-based collaborative-filtering system relies on a k-nearest-neighbor (KNN) algorithm [8] that identifies for each user, the k most similar other users according to a similarity metric. We consider the commonly used similarity metrics in user-based collaborative filtering. After identifying u's neighbors, the system ranks the items these neighbors have rated and recommends to u the top-ranking ones to which she has not yet been exposed.

We consider an adversary targeting a single user and which aims to extract information from the recommender [12]. Like [12], we assume the adversary has access to a subset of the target user's ratings, which we name *auxiliary information*. As described by [12], the adversary can obtain auxiliary information about the target in several ways. Many websites like Last.fm offer publicly available information about the browsing history of a user. Others, like Amazon.com, interface with social networks to post information like "I just bought item X". But an even more relevant threat arises for SMEs that offer recommendation services to multiple clients over different domains. Consider two client companies, for example online sport shops, that exploit the company's recommendation service. Each sport shop knows the ratings of its own customers on the items it sells, but it may be interested in knowing which items sold by the competitor are interesting for its own customers. The sport shop may thus play the role of an attacker whose auxiliary information consists of its own subset of the profile of a given customer. Existing work [14] studies this kind of threat in the context of targeted advertisement.

The adversary has also the ability to create a number of fake identities (i.e., Sybils). In the real world, this may be made complex by the need to purchase items, confirm accounts, or perform other costly actions to fill Sybil profiles. In this paper, we simply assume that the adversary can give each of its fake identities a user-profile consisting of a subset of items and ratings associated with the target. Finally, we also assume that the adversary knows the value of k (i.e., the size

of neighborhoods in the KNN algorithm) which allows her to create an appropriate number of Sybils.

We consider an attack successful when (i) Sybil nodes manage to obtain neighborhoods that consist only of the target node and other Sybil nodes, and/or (ii) they receive recommendations for items that appear in the target profile.

IV. Experimental Protocol

We implemented the Sybil attack on top of Mahout [2], a popular machine-learning framework developed by the Apache Foundation. This allows us to analyze the Sybil attack on a state-of-the-art user-based collaborative-filtering system.

In order to effectively implement the attacks, we slightly modified Mahout to model the behavior of Sybil users. Our modified Mahout implementation, as well as the code, and the scripts for running our experiments are publicly available [1]. We used publicly available anonymized real datasets exempt form ethical concerns.

A. Similarity Metrics

Section V evaluates the attack on seven similarity measures: the well-known cosine-similarity metric in three variants, two variants of the similarity measure from [9], the Jaccard index, and the Pearson correlation coefficient [40]. Cosine similarity reflects the similarity between two profiles (i.e. vectors of tuples representing the scores associated with items) by measuring the cosine of the angle between them. Jaccard considers the size of the intersection divided by the size of the union of two profiles, regardless the score associated with items. Finally, Pearson correlation consists of the covariance of the two profiles divided by the product of their standard deviations and turns out to be equivalent to a cosine similarity applied to the profiles obtained after centering the scores around their averages. Pearson correlation produces outcomes between -1 and $+1$ inclusive, while Cosine and Jaccard give values in $[0; +1]$. Their formal definition are available in [17].

We also consider two variants of cosine similarity: *Cos-overlap* and *CosineAvg*. The former computes the norms of the two profiles by counting only the items that are common to both of them while the latter uses the average rating of a user for non-rated items.

$$\textit{Cos-overlap}(u,n) = \frac{r_u \cdot r_n}{\sqrt{\sum_{i \in I_{u,n}} (r_{u,i})^2} \times \sqrt{\sum_{i \in I_{u,n}} (r_{n,i})^2}},$$

$$\textit{CosineAvg}(u,n) = \frac{\sum_{i \in I_u \cup I_n} \tilde{r}_{u,i} \times \tilde{r}_{n,i}}{\|\tilde{r}_u\| \|\tilde{r}_n\|} \quad ; \quad \tilde{r}_{x,i} = \begin{cases} r_{x,i} & \text{if } i \in I_x \\ \overline{r_x} & \text{if } i \notin I_x \end{cases}$$

where $r_{u,i}$ (resp. $r_{n,i}$) indicates the rating of user u (resp. n) on item i, $\overline{r_u}$ (resp. $\overline{r_n}$) the average of user u's (resp. n's) ratings, and $I_{u,n}$ the set of items rated by both u and n.

Finally, we consider two variants of the asymmetric similarity measure proposed in [9]: *WUP-u* and *WUP-n*, from the name of the system in [9]. These restrict their attention to the items rated by both users for the scalar product and for the norm of one of the two profiles depending on the variant. Given a user u and a potential neighbor n, the first variant, *WUP-u*, considers only the items rated by u that are also rated by n ($I_{u,n}$), and all the items rated by n (I_n).

$$\textit{WUP-u}(u,n) = \frac{\sum_{i \in I_{u,n}} r_{u,i} \times r_{n,i}}{\sqrt{\sum_{i \in I_{u,n}} (r_{u,i})^2} \times \sqrt{\sum_{i \in I_n} (r_{n,i})^2}},$$

The second variant, *WUP-n*, considers all the items rated by u (I_u) and only the items rated by n that also are also rated by u ($I_{u,n}$) and can be obtained by exchanging u and n in the above. Finally, we introduce our novel metric in Section VI and evaluate the attack on it in Section VII.

B. Datasets

We ran our experiments on the three datasets listed in Table I. ML-100k (called ML-1 in the following) is a trace from the MovieLens [22] online movie-recommendation service. Jester-1-1 is a trace from the Jester [19] online joke-recommendation service. It is the first third of Jester's dataset-1. MovieTweetings (MT) is a collection of movie ratings from IMDb users expressed as well-structured tweets [3]. We use the three parts of the dataset as one, and make our own splits where needed. We remove 23 erroneous ratings as they do not respect the expected 10-star rating format. Consequently, this also removes 3 users having only such erroneous ratings.

	# users	# items	# ratings	rating type
ML-1	943	1,682	100,000	$[1:5]$
Jester-1-1	24,983	100	1,810,455	$[-10.0:10.0]$
MovieTweetings	24,921	15,142	212,835	$[0:10]$

TABLE I: Characteristics of the datasets in terms of number of users, number of items, number of ratings and rating type.

ML-1 and Jester only contain information on subsets of users who rated at least 20 and 36 items respectively, which makes them less sparse than most real-world datasets. To investigate the impact of sparsity, we derive a number of variants from ML-1 with different levels of sparsity. Starting from ML-1, we progressively remove randomly chosen ratings and obtain four additional datasets (numbered ML-2 to ML-5). Table II details the number of ratings and the sparsity, expressed as the percentage of missing ratings in the user-item matrix, for each variant. For space reasons, we present plots only for ML and MT, and only discuss Jester in the text.

Dataset	Ratings	Sparsity
ML-1	100,000	93.69%
ML-2	50,351	96.82%
ML-3	25,180	98.41%
ML-4	13,698	99.14%
ML-5	7,621	99.52%
Jester-1-1	1,810,455	27.53%
MovieTweetings	212,835	99.94%

TABLE II: Sparsity of the original datasets and ML variants.

C. Auxiliary Information

The auxiliary information available to the adversary consists of a list of items and the associated ratings expressed by the target. We consider configurations with varying percentages of the target profile as auxiliary information. We observe that this is a relatively strong assumption on the prior knowledge held by the attacker. But it corresponds to the case of ratings available to a company in a cross-company recommendation system. In other cases, external available information is much less precise. For example, social networks may publish updates such as "Tom just saw Highlander" without specifying a rating.

The goal of the adversary consists in determining whether the target user liked a particular item, i.e. gave it a high-enough rating (≥ 3 in ML, ≥ 0 in Jester, and ≥ 6 in MT).

D. Assessment Metrics

We assess the Sybil attack according to three metrics. The first evaluates the ability of Sybils to build the ideal neighborhood associated with the target to carry out the attack. As discussed earlier, this consists of the target user and $k-1$ Sybils. More precisely, we measure the fraction of Sybils that obtain such an ideal neighborhood. The two other metrics, *yield* and *accuracy* evaluate the outcome of the attack. Each Sybil attacking the target receives 5 recommendations from its neighborhood, consisting of the items that receive the highest predicted rating $\hat{u_i}$, according to the following formula.

$$\hat{r}_{s,i} = \overline{r_s} + \frac{\sum\limits_{n \in N_s} (r_{n,i} - \overline{r_n}) \mathit{Sim}(s,n)}{\sum\limits_{n \in N_s} |\mathit{Sim}(s,n)|},$$

N_s being the nearest neighbors of Sybil s. We then define *yield* as the number of distinct items recommended to a given Sybil, and *accuracy* as the fraction of these items that actually exist in the target's profile.

Finally, we measure recommendation quality using the Root Mean Square Error (RMSE). To this end, we run 10-fold cross validation with a 90%-10% split. We build the KNN of each user on the training set and we issue recommendations to predict the ratings in the testing set. The RMSE measures how close recommendations are to the actual ratings of users in the testing set. For a given user u, it is thus defined as follows.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{r}_{u,i} - r_{u,i})^2}{n}} \quad ; \quad \hat{r}_{u,i} = \frac{\sum_{n \in N_u} \mathit{Sim}(u,n) * r_{n,i}}{\sum_{n \in N_u} \mathit{Sim}(u,n)}$$

V. Results

In this section, we analyze how similarity measures impact the success of the attack. For instance, a measure which drastically segregates users even with few changes in their profiles leaves enough room for the adversary to place its Sybil users around the target. In contrast, a similarity measure that does not differentiate among a large set of users will tend to provide similar scores to a number of potential neighbors in the KNN structure. This will make it more difficult for the adversary to create Sybil users that have only the target and other Sybils as potential neighbors.

To illustrate these differences among similarities, we computed the Complementary Cumulative Distribution Function (CCDF) of the similarity between all pairs of users for different similarity measures. Results (not shown for space reasons) suggest that the behaviors of the similarity measures can vary considerably in ML-1. The MovieTweetings dataset also exhibits significant differences between similarities, while in Jester, all similarities except Jaccard exhibit similar behaviors. The information in Table II, suggests that this difference results from the high density of Jester, with a sparsity value of only 27.53%, as opposed to values above 90% for ML-1 and MovieTweetings.

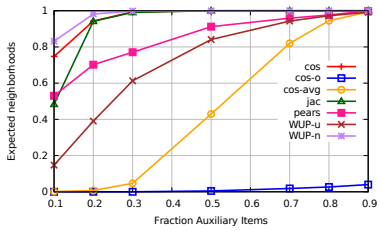
A. Similarity Metrics vs Sybil Attacks

We now start evaluating the effectiveness of the Sybil attack on recommenders based on the seven similarity metrics described in Section IV-D. Our results show that the attack succeeds to varying degree except with *Cos-overlap*. For these experiments, the auxiliary items available to the attacker consist of a random subset of the target's profile.

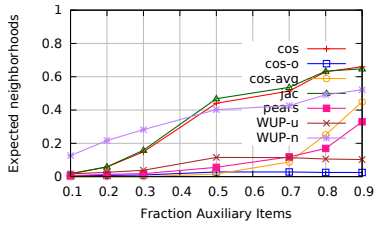
Figures 1 through 3 highlight a high variability of the attack's effectiveness. Figure 1 shows the percentage of sybil nodes that obtain their desired neighborhoods as a function of the fraction of auxiliary items in their possession. In Figure 1a, we can distinguish three groups of metrics. For the first group—*Cosine*, *Jaccard*, and *WUP-n*—knowing 20% of the items in the target profile allows each Sybil node to obtain a neighborhood that consists exactly of the target and $k-1$ other Sybils. The second group—*Pearson*, *WUP-u*, and *CosineAvg*—require instead as many as 80% or 90% of the target's items in order to achieve the same result. When knowing 30% of the items, *Pearson* and *WUP-u* only allow respectively 80% and 60% of the Sybils to build their target neighborhoods, while *CosineAvg* only provides the desired neighborhood to about 5% of the Sybils. Finally, the third group consists only of *Cos-overlap*: with this metric, only a few Sybils manage to obtain an ideal neighborhood even when they have as much as 90% of the target items in their profiles.

Figure 1b shows similar relative success rates, even though with lower absolute values. The first group, *Cosine*, *Jaccard*, and *WUP-n* allows Sybils to obtain ideal neighborhoods in the largest number of cases. The second group is slightly more resilient to the attack, while *Cos-overlap* makes it very hard for Sybils to obtain their ideal neighborhoods. In Jester, not shown for space reasons, the attack succeeds even with *Cos-overlap*, while *Jaccard* yields the worst attack performance. The reason for this behavior lies in the high density of Jester.

Figures 2 and 3 provide a different perspective on these results by depicting respectively the yield and accuracy obtained by the attack as a function of the fraction of the target's profile available as auxiliary items. To measure these, we have each Sybil request 5 recommendations from its neighborhood. With 10 Sybils this gives a maximum possible yield of 50. For ML-1 and MovieTweetings (Figures 2a and 2b), only *Cos-overlap*

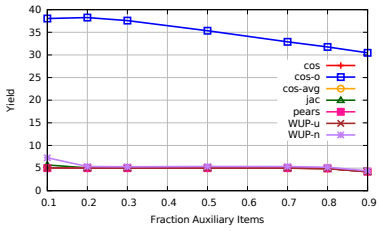


(a) ML-1

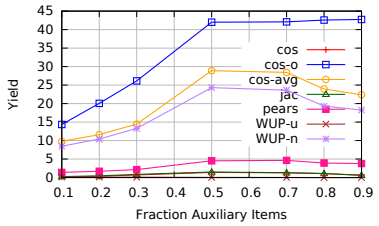


(b) MovieTweetings

Fig. 1: Fraction of Sybils with ideal neighborhoods for an isolated attack.

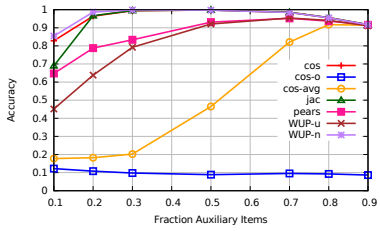


(a) ML-1

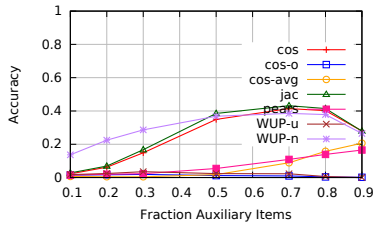


(b) MovieTweetings

Fig. 2: Yield for an isolated attack.



(a) ML-1



(b) MovieTweetings

Fig. 3: Accuracy for an isolated attack.

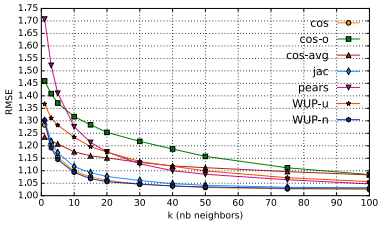
obtains a high yield. In Figure 2a all the remaining metrics achieve a yield of around 5%, while in Figure 2b yield values range from 0 to 25%.

Figure 3 shows that yield negatively correlates with accuracy. In Figures 3a and 3b, *Cos-overlap* allows Sybil nodes to obtain good predictions for a very small number of items (only 10% in ML-1 and close to 0 in MovieTweetings). The remaining lines in each plot follow the trend of Figure 1 pretty closely. In all three plots, Sybils obtain higher accuracy as the fraction of auxiliary items increase. With very high percentages of auxiliary items, accuracy decreases only because there profiles contain fewer and fewer items to guess. Finally, we observe that in Jester (not shown) Sybils obtain very high accuracy for pretty all metrics due to the high dataset density.

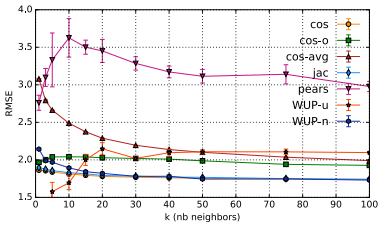
B. Recommendation Quality

We continue our analysis by showing that, for all the considered metrics, recommendation quality positively correlates with the effectiveness of the attack. To this end, Figure 4 depicts the root mean squared error (RMSE) obtained with each metric for varying neighborhood sizes in each of the datasets. Lower values mean better recommendations.

The plots show significant variability in performance depending on the metric and on the dataset. For ML-1, as in earlier plots, we can identify three groups of similarity metrics characterized by increasing levels of recommendation performance. The first group consists only of *Cos-overlap*. The good resilience to censorship exhibited by this metric in Section V-A comes at the cost of much poorer recommendation quality. By considering only the items that belong to both profiles being considered, *Cos-overlap* completely ignores the important distinction between users with specific—and thus useful in terms of recommendation—interests, and *hubs* with very unspecific behaviors.



(a) ML-1

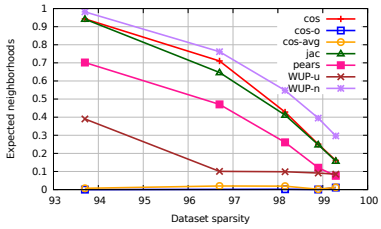


(b) MovieTweetings

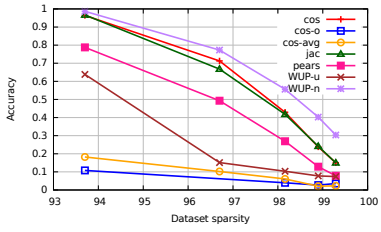
Fig. 4: RMSE using 10-fold cross validation.

The second group of metrics, consisting of *Pearson*, *WUP-u*, and *CosineAvg*, exhibits significantly better performance with a mean absolute error of about 0.85 and an RMSE of 1.1 with $k = 50$ neighbors. But the best results remain those of the last group of metrics: *Cosine*, *Jaccard*, and *WUP-n*.

In MovieTweetings, the relative differences between the metrics vary. *Pearson* exhibits particularly poor performance, *Jaccard* performs worse in Jester than in other datasets, while other metrics tend to reflect the relative differences highlighted in Figure 3. But, apart from any metric-specific remark, the main information we can extract from Figure 4 consists of its correlation with the data in Figures 1 through 3. For all metrics, high resistance to Sybil attacks results in poorer recommendation quality. This raises the research question of how to design a metric that can combine both benefits.

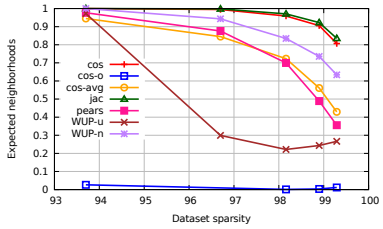


(a) Expected neighborhoods

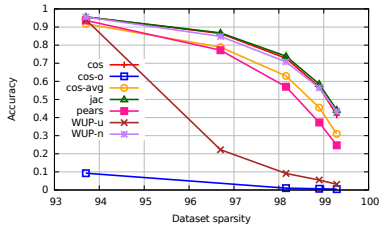


(b) Accuracy

Fig. 5: Fraction of Sybils with their expected neighborhood, yield, and accuracy (20% of auxiliary items, ML-1).



(a) Expected neighborhoods



(b) Accuracy

Fig. 6: Fraction of Sybils with their expected neighborhood, yield, and accuracy (80% of auxiliary items, ML-1).

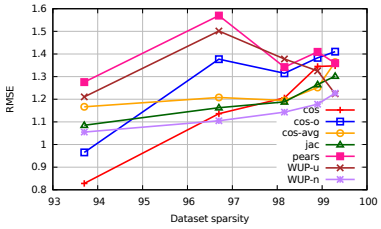


Fig. 7: RMSE versus sparsity with neighborhoods of size 15.

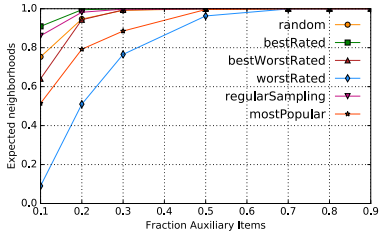


Fig. 8: Sybil attack with cosine similiary metric, ML-1 dataset and several auxiliary information scenarios.

C. Impact of Sparsity

We have already observed how the results in our different datasets reflect their different sparsity levels. We confirm this hypothesis, by considering five variants of the ML-1 dataset with increasing levels of sparsity.

Figures 5a and 5b depict, respectively, the proportion of expected Sybil neighborhoods and the attack's accuracy with varying sparsity levels and with 20% of auxiliary items. The figures show that the attack becomes less and less effective as sparsity increases. The fraction of expected neighborhoods drops drastically even for the metrics that tend to facilitate the attack in the base dataset (ML-1). Accuracy (Figure 5b) exhibits a similar behavior.

Figures 6a and 6b present the same analysis but with Sybils equipped with 80% of the target's profile. In this case, sparsity has a weaker impact on the attack, but accuracy still decreases with all of the metrics. Finally, Figure 7 shows that recommendation quality worsens with sparsity. This confirms the negative correlation between recommendation quality and accuracy highlighted in Section V-B.

D. Choice of Auxiliary Information

We now examine the information impacts the success of the attack. To this end, we consider *Cosine*, as it is the most widely used metric due to its good recommendation performance, and see how the attack behaves in the following six scenarios for the choice of auxiliary information. Let t be the target profile sorted by rating, and let $\mathcal{A}$ux be the set of auxiliary items and ratings. We consider six scenario: 1) *Random* where $\mathcal{A}$ux constitutes a random subset of t, 2) *BestRated* where $\mathcal{A}$ux comprises the best rated items in t, 3) *WorstRated* where $\mathcal{A}$ux comprises the worst-rated items in t, 4) *BestWorstRated* where half of $\mathcal{A}$ux comprises the best-rated items in t and the other half the worst rated, 5) *RegularSampling* where $\mathcal{A}$ux consists of items picked at regular intervals from t, 6) *MostPopular* where $\mathcal{A}$ux contains items from t that are also the most popular in the whole recommender. Based on these six scenario we can show the following lemma (proof is provided in the appendix).

Lemma 1. *Let $\mathcal{A}ux_{scenario}$ be a set of auxiliary items in a given scenario. We consider $s_{\mathcal{A}ux}$ a Sybil user whose profile only consists of auxiliary information $\mathcal{A}ux_{scenario}$. The BestRated scenario maximizes the Cosine similarity of the Sybil with the target, i.e.*

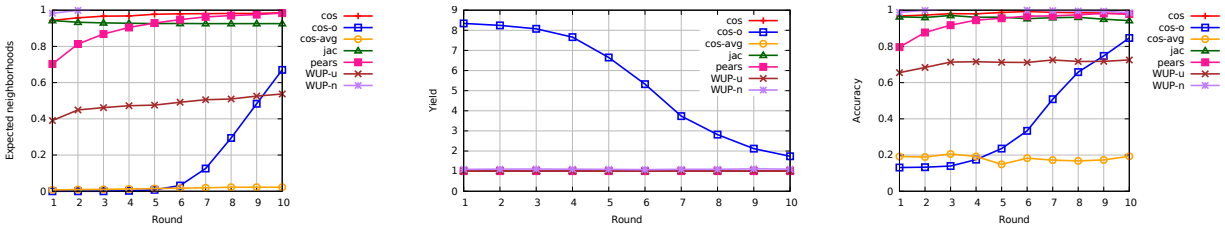
$$\mathcal{A}ux_{bestRated} = \arg\max_{\mathcal{A}ux \subset t} [cos(s_{\mathcal{A}ux}, t)]$$

Proof. The profile of the Sybil user is a subset of the target's profile. For each item in $\mathcal{A}$ux, the ratings are the same in both profiles because of the attack definition. Thus, we obtain the following:

$$cos(s_{\mathcal{A}\text{ux}}, t) = \frac{\sum s_i^2}{\|t\| \times \sqrt{\sum s_i^2}} = \frac{\sqrt{\sum s_i^2}}{\|t\|}$$

$cos(s_{\mathcal{A}\text{ux}}, t)$ is maximised by the bestRated scenario. □

To complement this lemma, we also ran experiments on the ML-1 dataset: Figure 8 depicts the result. As expected BestRated maximizes the success of the attack, but Random, which we used in the previous sections, and RegularSampling perform almost as well. MostPopular achieves poorer performance, possibly because popular items are less discriminating, while WorstRated performs the worst.



(a) Expected Neighborhoods (b) Yield (c) Accuracy

Fig. 9: Adaptive Sybils on the ML-1 dataset. Sybils have 20% of auxiliary items.

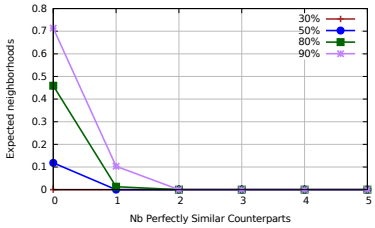


Fig. 10: Attack performance with ML-1: ideal neighborhoods

E. Adaptive Sybils with Standard Metrics

We have so far considered a single round of attack. In a real setting, however, an attacker may iterate the attack while incorporating the new information he/she learned about the target. In Figure 9, we evaluate exactly this scenario in the ML1 dataset. Each Sybil requests one recommendation at a time. Once the Sybil receives the recommendation, we add the recommended item to the Sybil's profile with the score it has in the profile of the user that provided the recommendation. We then take this new item into account to compute a new neighborhood and obtain a new round of recommendations. We repeat the process for 10 rounds.

Results shows that, with most of the metrics, the performance of the attack remains constant throughout the recommendation rounds. However, *Cos-overlap* exhibits a dramatic increase in both the fraction of expected neighborhoods and accuracy, with a corresponding decrease in yield. The reason lies in the way *Cos-overlap* treats the items that appear in the Sybil's profile but not in that of the target, and on the assumptions we made on the attacker. When a Sybil receives a recommendation, this may come either from the target's profile, or from some other user's profile. However, *Cos-overlap* only considers items from the target user's profile. As a result, when a Sybil receives a recommendation that is not in the target profile, this does not penalize its similarity with the target. However, when it receives a recommendation for an item that is both in the profile of the target and in that of another user, our assumption that the Sybil can guess the profile of the target penalizes its similarity with the other user. This explains the increase in accuracy over successive rounds in the case of *Cos-overlap*.

VI. Towards Sybil Resistance

Our analysis of the results in Sections V-A and V-B reveals a very clear trade-off between the quality of recommendation and Sybil resistance. In this section, we propose a new metric that minimizes the impact of this trade-off. To this end, we examine the peculiar performance of *Cos-overlap* on ML-1 and extract some guidelines for the design of our new metric.

A. Understanding Cosine-Overlap

The denominator of *Cosine* similarity discounts the scores of users with very large profiles thereby benefiting those that have more specific interests. But *Cos-overlap* entirely removes this behavior and considers only the ratings of items that appear in both user profiles and completely ignores those that appear in only one of them. This means that two users may have a similarity of 1 even if their item sets differ significantly. We indeed notice that a very large proportion of users have perfectly similar counterparts (i.e. other users with whom they have a similarity of 1) from the point of view of *Cos-overlap*.

The poor discriminatory power of *Cos-overlap* makes it hard for Sybils to distinguish the target and the other Sybils from the target's perfectly similar alter-egos. Figure 10 breaks down the data from Figure 1 and shows the fraction of perfect neighborhoods as a function of the number of perfectly similar counterparts of the target. Each line corresponds to a different amount of auxiliary knowledge made available to the attacker. The data for users that have no perfectly similar counterparts pretty much follows the behaviors of the other similarity metrics in Figure 1. The percentage of Sybils that get an ideal neighborhood strongly depends on the amount of available auxiliary knowledge. However, as soon as the target has at least 2 perfectly similar counterparts, none of the Sybils manages to obtain an ideal neighborhood, regardless of the amount of auxiliary knowledge they have. The results for prediction accuracy (not shown for space reasons) follow a similar pattern. Sybils can guess the profiles of users that are sufficiently unique, but can do little for users that have good alter-egos.

While the presence of perfectly similar counterparts constitutes an asset for Sybil resistance, it clearly hampers the system's ability to provide good recommendations. Consider a user, A, with two perfectly similar alter egos, B, and C. A and B share a single common rating on a single common item. A and C, on the other hand, share common ratings on a significant portion of their two profiles. Clearly, C will be a better candidate than B to provide recommendations to A. But *Cos-overlap* will consider B and C as equally good.

B. Two-step Similarity Metric

The above observations suggest that a metric should, on the one hand, discriminate good from bad profiles for recommendation, while, on the other, prevent Sybils from identifying the target and other Sybils.

We satisfy these requirements with *two-step*, a novel similarity metric with a composite structure. Given a user u, and a potential neighbor n, the first step may employ any existing similarity metric (e.g. *Cosine*), and post-processes its values so that all the potential neighbors that score beyond a certain threshold appear to be the same to u. Unlike focusing on a small subset of items as in the case of *Cos-overlap*, using a threshold makes it possible to coalesce users that are likely to provide similar results in terms of recommendation. This makes it difficult for a Sybil user to obtain a neighborhood that contains the desired target user.

The second step of the metric goes beyond the threshold and attempts to distinguish which of the top-scoring potential neighbors (those with a first step above the threshold) may be useful to compute recommendations for user u. The recommendation process consists in finding potentially interesting items in the profiles of u's neighbors. This implies that a neighbor that has no items that do not appear in u's profile brings nothing to the recommender and should therefore be discarded. Rather, a good neighbor should have at least some items that do not appear in u's profile.

The second step therefore differentiates the potential neighbors that score above the threshold by taking into account the number of items in their profiles that do not appear in the profile of u. Because the recommender computes the neighborhoods for all users, including the Sybils, this heuristic has the beneficial effect of discouraging the presence of other Sybils in the neighborhood of a Sybil user, thereby making the attack more difficult.

To summarize, the threshold makes it hard for a Sybil to differentiate the target, or another Sybil from other very similar nodes. The second step complements this feature by *preferring* legitimate users to Sybils. In the following, we describe the details of our two-step metric.

C. Two-step details

Let u be a user for which we have to evaluate the goodness of w as a neighbor. Both, u and w may be legitimate users or Sybils. Also, with some abuse of notation, let $w - u$ denote the set of items that appear in w's profile but not in u's.

Let *Sim* be a similarity metric, for example *Cosine*. In the first of the two steps, we compute the similarity between u and w, *Sim*(u, w). If *Sim*(u, w) is less than a threshold th_u then we use *Sim*(u, w) as the final similarity value. Otherwise we compute $th_u + f_{i,u}(|w - u|)$, where (i) $|w - u|$ is the number of items that appear in the profile of w but not in that of u, (ii) i is the total number of items in the system, and (iii), $f_{i,u} : \mathbb{N} \to [0, 1 - th_u]$ is a function defined as follows.

$$f_{i,u}(x) = (1 - th_u)\frac{x}{i}$$

This increasing function attempts to ensure that the neighbor's profile contains some items that are not in u's profile. By combining the two above steps, we obtain the following definition for our two-step metric.

$$2{-}step(u, v) = \begin{cases} \textit{Sim}(u, v) & \text{if } \textit{Sim}(u, v) < th_u \\ th_u + f_{i,u}(|v - u|) & \text{if } \textit{Sim}(u, v) \geq th_u \end{cases}$$

VII. Evaluating Two-step

We evaluate $2{-}step$ using the same metrics as in Section V. To set the metric's threshold, we make a pass on the entire user-item matrix before computing the nearest-neighbor graph. For each user u, we compute the similarity of u with all the other users. We then round similarity values to the nearest hundredth, sort them, remove all duplicate values, and set the threshold as the t^{th} percentile of the resulting sorted sequence: we considered values of $t \in \{80, 90\}$. Figure 11 shows the RMSE score with ML-1. We have the same recommendation quality as with the cosine metric and significantly better recommendation quality than *Cos-overlap*. This is the first advantage of this metric: we ensure a very good recommendation quality, in terms of RMSE.

A. Basic Attack on Two-Step

Next, we evaluate how effectively $2{-}step$ protects users from Sybil attacks. To this end, we first consider the attack from [12] we analyzed in previous sections. Figure 12 depicts its performance in terms of expected neighborhood and accuracy. The plot shows that none of the Sybils manages to obtain its expected neighborhood, while accuracy never exceeds 25%. Even this small number of correct guesses does not result from a truly successful attack, but from the presence of other nodes whose profiles resemble that of the target.

The plot also shows that accuracy decreases with the number of auxiliary items. This may appear counter-intuitive, but as the proportion of auxiliary items increases, these start to include more and more of the items that the target shares with its neighbors. Since the attacker never manages to have the target as a neighbor, the number of possible correct guesses based on other nodes' profiles decreases rapidly.

Although the above results appear very promising for $2{-}step$, the basic attack does not specifically target this metric. In the following, we therefore extend the attack in order to evaluate the resistance of $2{-}step$ in the worst case.

B. An Attack Targeting Two-Step

The success of the Sybil attack relies on the ability (i) to gather all other Sybils in the attacker's neighborhood, and (ii) to have the target as a neighbor. In a system based on *Cosine*, giving all Sybils the same profile maximizes the chances of satisfying (i), while giving them as many items as possible from the target profile maximizes those of satisfying (ii).

In the case of $2{-}step$, on the other hand, having the same profile does not suffice to satisfy (i), and gathering as many auxiliary items as possible can offer only limited benefits with

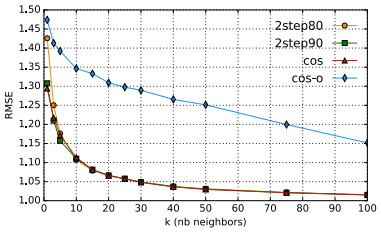


Fig. 11: RMSE, using 10-fold cross validation (ML-1).

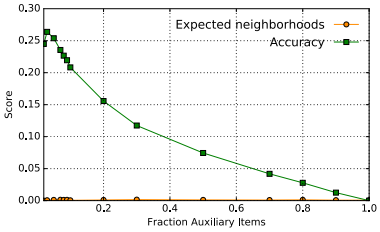


Fig. 12: Fraction of Sybils with their expected neighborhood and accuracy (standard attack, $2-step$, ML-1).

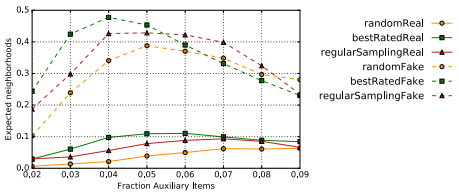
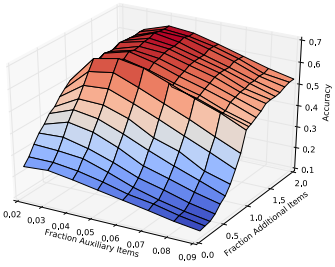
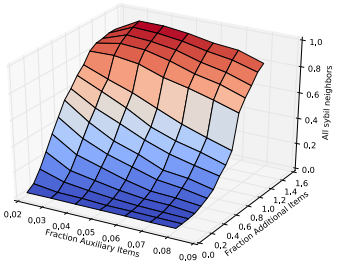


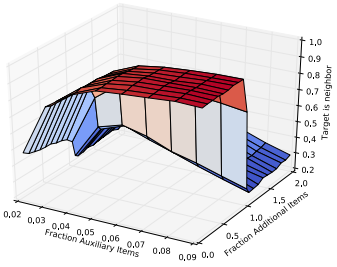
Fig. 13: Expected neighborhoods score, with real and fake additional items (ML-1).



(a) Accuracy



(b) All other Sybils are neighbors



(c) Target is Neighbor

Fig. 14: Sybil attack with ML-1 dataset, $2-step$ 80 similarity metric, bestRated scenario, and fake additional items

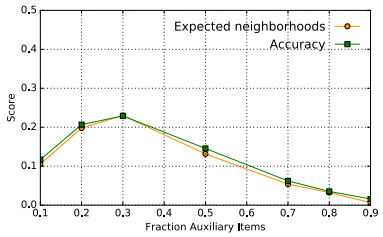
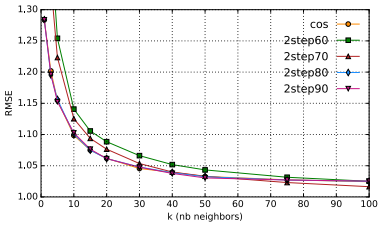
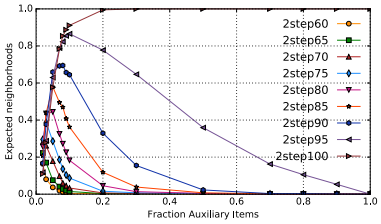


Fig. 15: Sybil attack for $2-step$ 80, bestRated scenario, and fake additional items (Movie Tweetings).



(a) 10-fold RMSE



(b) Expected Neighborhoods

Fig. 16: Threshold Analysis with ML-1 dataset, bestRated scenario, and fake additional items.

respect to (ii). To obtain a high similarity value with $2-step$, two users (real or Sybil users) must have a large enough common set of items, and a large enough non-shared set.

To explicitly target our $2-step$ metric, we therefore modify the basic attack from [12] by considering both of these two requirements. This yields a $2-step$-specific attack with two parameters: a set auxiliary items taken from the target's profile, and a set of additional items. The presence of these two parameters, however, makes it more difficult for the attacker to identify a winning strategy. It becomes reasonable to wonder whether a strategy independent of the dataset actually exists.

To answer this question, we carry out a worst case analysis by varying both parameters through a range of values. To begin with, we consider the possible strategies for choosing auxiliary and additional items. Figure 13 shows the fraction of expected neighborhoods among the Sybils using Sybil profiles of the same size as the target profile, and increasing fractions of auxiliary items, for different choice strategies.

With respect to auxiliary items, the plot confirms BestRated as the best option for the attacker, and so the worst case scenario for the algorithm. The rest of the plot confirms the results of Figure 8, with RegularSampling, and Random performing slightly worse than best rated. For clarity, we do not show the remaining worse performing strategies.

With respect to additional items, the plot shows two strategies: *real* and *fake*. With the former each Sybil chooses a set of *real* items from the systems that are not in the target's profile and that have not been chosen by any other Sybil. With the latter, each Sybil generates and chooses a set of *fake* items. *Fake* consistently achieves better performance because they cannot be shared with any other node (Sybil or legitimate). This guarantees the maximum bonus for each Sybil node in the computation of $2-step$.

Finally, the plot also shows an important difference with respect to Figure 8. With *Cosine*, the performance of the attack increases with the fraction of auxiliary items. But with $2-step$,

this no longer happens. Rather, each of the curves exhibits a different ideal fraction of auxiliary items, and all such ideal fractions correspond to very low values. To understand this behavior, Section VII-C deepens our analysis by searching for a worst-case set of parameters.

C. Searching for a Worst Case Scenario

To identify a worst-case scenario for $2-step$, we consider the best configuration for the attacker: best-rated auxiliary items and fake additional items. We experimentally search for the worst-case scenario by varying the sizes of the sets of auxiliary and additional items: respectively $\mathcal{A}$ux, and $\mathcal{A}$dd. Since the best number of additional items likely depends on the size of the target profile, we represent this number as a fraction. Let t be the size of the target profile, p the fraction of auxiliary items. We define the fraction q of additional items as $q = \frac{\mathcal{A}\text{dd}}{(1-p)|t|}$. A fraction $q = 1$ therefore gives Sybils a profiles that has the same size as that of the target node.

Figure 14 shows the performance of the attack with varying values of both p and q in the form of 3D plots. Results confirm the difficulty of finding a very successful attack configuration on $2-step$. Accuracy never exceeds 60% in the same dataset that yielded close to 100% accuracy in Figure 3a. As already shown in Figure 13, this 60% value is achieved for a very small value of p. But Figure 14a, shows that accuracy increases until $q = 1$ and then plateaus. The plot shows values of $q \leq 2$, but we experimentally verified that accuracy further decreases for larger values of q with values below 0.3 for $q \geq 5$.

To clarify this behavior, Figures 14c and 14b show the distribution of the two conditions required to achieve a perfect neighborhood: having the target as a neighbor, and having all Sybils as neighbors. Figure 14b shows that increasing q always increases the probability to have all other Sybils as neighbors, while p exhibits the same point of maximum as in Figure 14a. On the other hand, the probability of having the target as a neighbor drops significantly when $q \geq 1$. This explains the plateauing accuracy values in Figure 14a for $q \geq 1$.

D. Attack Performance across Datasets

We now examine the effectiveness of the $2-step$-specific attack on MT and Jester. Figure 15 depicts the results obtained on MT with a fraction of additional items, $q = 1$. The curves appear similar to those on the ML-1 dataset, but with a lower peak. At most, 25% of Sybil users obtain the ideal neighborhood, with a fraction of auxiliary items $p = 0.3$ (i.e., 30%). The plot also shows that the score for accuracy closely mimics that for the fraction of *Expected Neighborhoods*.

The results for Jester, not shown for space reasons, show instead that none of the Sybils obtain the expected neighborhood and that the target appears in the Sybil's neighborhood only in a very small fraction of cases (less than 1%). While this results correspond to an ineffective attack, accuracy still achieves very high values thanks to the density of the dataset. To test this conclusion, we also evaluated an attacker that selects k random users as neighbors, instead of a KNN search and even this completely random attack yielded high accuracy values.

Overall, the results for $2-step$ highlight the difficulty of identifying suitable parameters for a successful attack. The low scores make it hard for the attacker to know whether it is really observing a part of the target's profile.

E. Threshold Analysis

We have so far evaluated $2-step$ with two values of its percentile threshold. We now show that a wide range of values offers good recommendation quality while resisting to attacks. Figure 16a shows that 80% and 90% thresholds achieve the best recommendation performance in ML-1, giving as low an RMSE as cosine. Figure 16b confirms instead the expectation that the lower the threshold the worse the performance of the attack. This justifies the choice of an 80% threshold in Sections VII-A through VII-D.

Figure 17 completes our analysis of $2-step$'s threshold by showing the same plots for the MT dataset. Interestingly, a threshold of 70% performs equally well as cosine, while one of 60% performs even better for large neighborhood sizes.

F. Comparison with PPNS

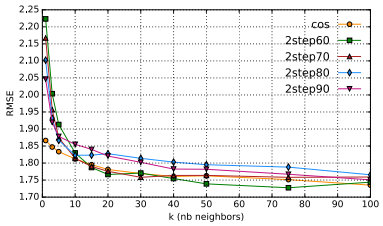
We now show that our similarity-based defense mechanism outperforms state-of-the-art techniques by comparing its attack resilience with that of PPNS [29]. PPNS seeks to protect user-based collaborative filtering from the same attack we consider in this paper by modifying the neighbor-selection process. For a given users, it partitions all other users in descending order of similarity in partitions of size k. It then extracts k neighbors from the top β partitions while maximizing recommendation accuracy subject to the constraint that at least one neighbor be extracted from the β-th one. The optimal solution consists in selecting $k-1$ neighbors from the first partition and 1 from the β-th one [29]. We therefore experimented with this strategy.

In terms of RMSE, results on the ML-1 datasets appear indistinguishable from those in Figure 4a and, with k Sybil users, the attacker never manages to have the target in a Sybil's neighborhood, making the attack unsuccessful. However, we also tested a stronger version of the attack with $\beta * k$ Sybils. In this case, the attacker can easily build neighborhoods that include not only Sybils but also the target node.

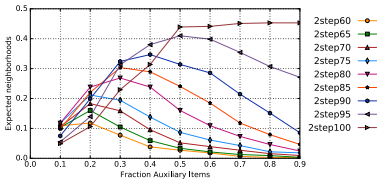
Figure 18 shows the results of the experiments with $\beta * k$ Sybils in terms of expected neighborhoods and attack accuracy. To provide a conservative comparison, we also perform an attack on 2-step using the specific attack with $\beta * k$ Sybils. The plot clearly shows that while 2-step provides practically the same protection as with k Sybils, PPNS becomes helpless with $\beta * k$ Sybil nodes. Since the difficulty in orchestrating an attack with a large number of Sybil users remains limited, this experiments shows the superiority of our approach.

G. Adaptive Sybils with 2-Step

Finally, Figure 19 shows the result of adaptive Sybils with $2-step$. The *Expected neighborhoods* score decreases over rounds. If the attacker includes a correct guess as an auxiliary item, then the Sybils will have a lower bonus at the second step during the following round.



(a) 10-fold RMSE



(b) Expected Neighborhoods

Fig. 17: Threshold Analysis with Movie Tweetings dataset, bestRated scenario, and fake additional items.

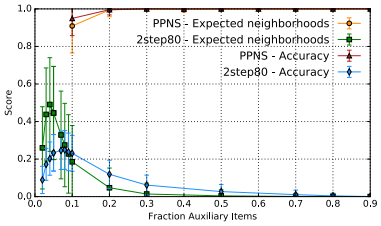


Fig. 18: Simple attack using PPNS (ML-1), compared with 2step80 specific attack (40 sybils).

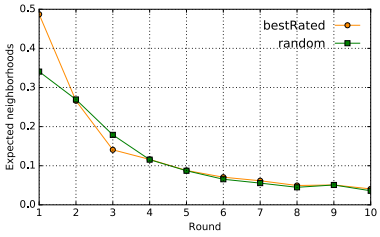


Fig. 19: Adaptive Sybil attack over 10 rounds with ML-1, $2-step$-80, 4% aux. information, and fake additional items

VIII. Theoretical Analysis

We conclude with a brief theoretical comparison of *Cosine* and $2-step$ in the context of binary ratings; an additional property of $2-step$ is available in [17].

Let t be a target user with $|t| > 1$ and let $\mathcal{S}$ be the set of all Sybil users generated by the attacker.

Assumptions 1. *Let t be a user such that: (i) No other user has the same profile as t; (ii) there exists no user u with $u \subset t$ and $|t-u| = 1$; and (iii) there exists no user u' with $|u'| = |t|$ and $|u' \cap t| = |t| - 1$.*

Theorem 1. *Let s_{aux} be a Sybil user with a profile consisting of the auxiliary information aux. Given Assumptions 1, there exists aux that ensures: $\forall u \notin \mathcal{S}$, $cos(s,t) > cos(s,u)$*

Theorem 1 confirms the vulnerability of *Cosine* similarity observed in Section V. The proof is available in [17]. The next theorem shows that $2-step$ effectively protects the recommender from the basic attack.

Theorem 2. *Given Assumptions 1, let aux be such that: $aux \in t$, $|aux| = |t| - 1$, and $\forall u \subset t \implies u \subset aux$. Let s_{aux} be the associated Sybil user.*
Let $N_v = \#\{u \mid cos(u,v) \geq th_v\}$ be the number of users above the threshold for user v.
Let $C_v = \#\{u \mid u \subset v\}$ be the number of users with a profile which is a subset of v's profile. Then, if $C_t < N_{s_{aux}}$, we have: $\exists u \notin \mathcal{S}$, $2-step(s_{aux},t) \leq 2-step(s_{aux},u)$

Proof. Among $N_{s_{aux}}$, we have at most k users that are a Sybil or the target. Among C_t, we have at least the k Sybils. Therefore, as $C_t < N_{s_{aux}}$, we can ensure there exists a user $u \notin \mathcal{S}$ above the threshold $th_{s_{aux}}$ which does not verify $u \subset t$. We have $cos(s_{aux},u) \geq th_{s_{aux}}$.

If $cos(s_{aux},t) < th_{s_{aux}}$ then we deduce the result.

Otherwise, because u is not a subset of t, we have $|t \cap u| < |u|$. There exists at least one item in u but not in t and thus not in s_{aux}. We have : $|u - s| \geq 1 = |t - s|$. We deduce: $2-step(s_{aux},t) \leq 2-step(s_{aux},u)$ □

Even if it concentrates on the basic attack, the above theorem provides a hint about the difficulties associated with targeting $2-step$. To generalize this result to the $2-step$-specific attack from Section VII-B, we collected, in Table III, the components of the computation of $2-step$. The goal of the attacker consists in maximizing the values of $2-step(s,t)$ and $2-step(s,s')$, while minimizing that of $2-step(s,u)$. By increasing the fraction of auxiliary items, the attacker can increase the *Cosine* component of both $2-step(s,t)$ and $2-step(s,u)$. The former will, in general, increase faster as $|aux \cap u| < |aux|$, but ultimately both get capped at the same value, th_s. If we look at the bonus column, on the other hand, we observe that an increase in aux causes a faster decreases in $2-step(s,t)$ than in $2-step(s,u)$. This means that too high values of aux will cause the target to exit from the attacker's neighborhood. This suggests that aux should be large enough to enable t to exceed the threshold but not too much to avoid too sharp a decrease in t's bonus.

A similar analysis holds for the number of additional items, add. By increasing it, the attacker can increase the bonus for $2-step(s,s')$, but it also reduces the corresponding *Cosine* component. Again, while a small increase can be beneficial, too large an increase will bring $cos(s,s')$ below the threshold, preventing Sybils from being in each other's neighborhood.

This reasoning shows that the best values of aux and add cannot be predetermined in advance, but depend on the specific dataset. This makes it particularly difficulty to attack $2-step$.

IX. Concluding Remarks

We presented a comprehensive experimental analysis of the impact of similarity metrics on the Sybil resilience of existing user-based collaborative-filtering systems. Our results, obtained on a state-of-the-art recommendation framework highlight the limits of existing similarity metrics. We addressed these limits by proposing a novel Sybil-resistant similarity metric. We showed that our novel metric not only

Similarity	Cosine part	Bonus
$2-step(s,t)$	$\frac{\lvert aux\rvert}{\sqrt{(\lvert aux\rvert+\lvert add\rvert)\lvert t\rvert}}$	$\propto \lvert t\rvert - \lvert aux\rvert$
$2-step(s,s')$	$\frac{\lvert aux\rvert}{\lvert aux\rvert+\lvert add\rvert}$	$\propto \lvert add\rvert$
$2-step(s,u)$	$\frac{\lvert aux\cap u\rvert}{\sqrt{(\lvert aux\rvert+\lvert add\rvert)\lvert u\rvert}}$	$\propto \lvert u\rvert - \lvert aux\cap u\rvert$

TABLE III: Components of $2-step$ similarity between relevant pairs of nodes. t is the target; s and s' are Sybils; and $u \in \mathrm{RS} \setminus \{s, s', t\}$.

resists to the basic attack designed for standard metrics, but also to a specific attack designed to exploit its very structure, while outperforming existing solutions.

Exploring variants of the attack, for instance in a system that relies on dissimilarity metrics or on an item-based approach, would constitute interesting avenues for future work.

X. Acknowledgements

This work was partially funded by the Region of Brittany, France, by the French ANR projects Pamela (ANR-16-CE23-0016-04), by the Profile project granted by the Labex Comin-Labs excellence laboratory (ANR-10-LABX-07-01), and by the Google Focused Research Award Web Alter-Ego.

References

[1] https://github.com/fdemoor/recopriv.
[2] "Mahout," https://mahout.apache.org/.
[3] "Movietweetings," http://2014.recsyschallenge.com/dataset/.
[4] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *PODS*. ACM, 2001.
[5] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *SIGMOD*. ACM, 2000.
[6] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci, "Enhancing privacy and preserving accuracy of a distributed collaborative filtering," in *RecSys*. ACM, 2007.
[7] S. Bhagat, U. Weinsberg, S. Ioannidis, and N. Taft, "Recommending with an agenda: Active learning of private attributes using matrix factorization," in *RecSys*. ACM, 2014.
[8] A. Boutet, A. M. Kermarrec, N. Mittal, and F. Taiani, "Being prepared in a sparse world: The case of knn graph construction," in *ICDE*, 2016, pp. 241–252.
[9] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec, "WHATSUP: A decentralized instant news recommender," in *IPDPS*. IEEE, 2013.
[10] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A. Kermarrec, "Privacy-preserving distributed collaborative filtering," in *NETYS*. Springer, 2014.
[11] A. Boutet, D. Frey, A. Jégou, A.-M. Kermarrec, and H. B. Ribeiro, "Freerec: An anonymous and distributed personalization architecture," in *NETYS*. Springer, 2013.
[12] J. Calandrino, A. Kilzer, A. Narayanan, E. Felten, and V. Shmatikov, ""you might also like:" privacy risks of collaborative filtering," in *SP*. IEEE, 2011.
[13] J. Canny, "Collaborative filtering with privacy via factor analysis," in *SIGIR*, 2002.
[14] C. Castelluccia, M.-A. Kaafar, and M.-D. Tran, "Betrayed by your ads!" in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science. Springer, 2012, vol. 7384.
[15] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *WWW '07*. ACM, 2007.
[16] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *WWW '11*. ACM, 2011, pp. 577–586.
[17] D. Frey, A. Boutet, F. De Moor, R. Guerraoui, A.-M. Kermarrec, and A. Rault, "Collaborative Filtering Under a Sybil Attack: Similarity Metrics do Matter!" Inria, Research Report, Apr. 2018. [Online]. Available: https://hal.inria.fr/hal-01767059
[18] D. Frey, R. Guerraoui, A.-M. Kermarrec, and A. Rault, "Collaborative filtering under a sybil attack: Analysis of a privacy threat," in *EuroSec*. ACM, 2015.
[19] K. Goldberg, T. Roeder, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, pp. 133–151, 2001.
[20] I. Gunes, A. Bilge, and H. Polat, "Shilling attacks against memory-based privacy-preserving recommendation algorithms," *TIIS*, vol. 7, no. 5, pp. 1272–1290, 2013.
[21] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg, "Eclipse attacks on bitcoin's peer-to-peer network," in *USENIX Security*, 2015, pp. 129–144.
[22] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *SIGIR*. ACM, 1999.
[23] T. R. Hoens, M. Blanton, and N. V. Chawla, "Reliable medical recommendation systems with patient," *IHI*, 2010.
[24] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *SIGMOD*. ACM, 2005.
[25] A. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. Lagendijk, and Q. Tang, "Privacy in recommender systems," in *Social Media Retrieval*, ser. Computer Communications and Networks. Springer, 2013, pp. 263–281.
[26] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *ICDM*. IEEE, 2003.
[27] D. Li, Q. Lv, H. Xia, L. Shang, T. Lu, and N. Gu, "Pistis: A privacy-preserving content recommender system for online social communities," in *WI-IAT*. IEEE, 2011.
[28] G. Linden, J. Jacobi, and E. Benson, "Collaborative recommendations using item-to-item similarity mappings," Jul. 24 2001, uS Patent 6,266,649.
[29] Z. Lu and H. Shen, "A security-assured accuracy-maximised privacy preserving collaborative filtering recommendation algorithm," in *IDEAS*, 2014, pp. 72–80.
[30] ——, "An accuracy-assured privacy-preserving recommender system for internet commerce," *CoRR*, vol. abs/1505.07897, 2015.
[31] A. Machanavajjhala, A. Korolova, and A. D. Sarma, "Personalized social recommendations: accurate or private," *VLDB*, 2011.
[32] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *SIGKDD*. ACM, 2009.
[33] A. Ozturk and H. Polat, "From existing trends to future trends in privacy-preserving collaborative filtering," *Wiley Int. Rev. Data Min. and Knowl. Disc.*, vol. 5, no. 6, pp. 276–291, Nov. 2015.
[34] R. Parameswaran and D. M. Blough, "Privacy preserving collaborative filtering using data obfuscation," in *Granular Computing, 2007. GRC 2007. IEEE International Conference on*. IEEE, 2007, pp. 380–380.
[35] H. Polat and W. Du, "Svd-based collaborative filtering with privacy," in *SAC*. ACM, 2005.
[36] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis, "Privacy risks in recommender systems," *IEEE Internet Computing*, vol. 5, no. 6, pp. 54–62, Nov. 2001.
[37] P. Resnick and R. Sami, "The influence limiter: Provably manipulation-resistant recommender systems," in *RecSys*, 2007, pp. 25–32.
[38] C. E. Seminario and D. C. Wilson, "Attacking item-based recommender systems with power items," in *RecSys*. ACM, 2014.
[39] A. Singh, T. wan johnny Ngan, P. Druschel, and D. S. Wallach, "Eclipse attacks on overlay networks: Threats and defenses," in *In INFOCOM*, 2006.
[40] P. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Addison Wesley, 2005.
[41] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "BlurMe: Inferring and obfuscating user gender based on ratings," in *RecSys*. ACM, 2012.
[42] X. Zhao, W. Chen, F. Yang, and Z. Liu, "Improving diversity of user-based two-step, recommendation algorithm with popularity normalization," in *DASFAA*, 2016, pp. 15–26.
[43] X. Zhao, Z. Niu, and W. Chen, "Interest before liking: Two-step recommendation approaches," *Knowledge-Based Systems*, vol. 48, pp. 46 – 56, 2013.
[44] T. Zhu, G. Li, Y. Ren, W. Zhou, and P. Xiong, "Differential privacy for neighborhood-based collaborative filtering," in *ASONAM*. ACM, 2013.